Modular Arithmetic Exploration

Katherine E. Stange, CU Boulder

Exponents, efficiently

We've seen that the summands and factors of large computations modulo n can be reduced before adding or multiplying, which saves time.

But we've also seen that exponents cannot be reduced.

The purpose of this exploration is to discover the most efficient way to compute large exponents.

Let's work modulo 100.

1. Compute the following with only *one multiplication each time*, at each stage possibly using previous results from the lines above the current line. You can just point out what you *would* multiply and not bother actually doing it.

   The point of this exercise is to pick up on tricks and patterns, so pay attention!

   The first two have been done for you.

   - $2^2 \pmod{100}$
     can be computed by multiplying $2 \pmod{100}$ and $2 \pmod{100}$

   - $2^4 \pmod{100}$
     can be computed by multiplying $2^2 \pmod{100}$ and $2^2 \pmod{100}$

   - $2^8 \pmod{100}$
     can be computed by multiplying $\quad \pmod{100}$ and $\quad \pmod{100}$

   - $2^{16} \pmod{100}$
     can be computed by multiplying $\quad \pmod{100}$ and $\quad \pmod{100}$

   - $2^{32} \pmod{100}$
     can be computed by multiplying $\quad \pmod{100}$ and $\quad \pmod{100}$

   - $2^{64} \pmod{100}$
     can be computed by multiplying $\quad \pmod{100}$ and $\quad \pmod{100}$

- $2^3 \pmod{100}$
  can be computed by multiplying (mod 100) and (mod 100)

- $2^7 \pmod{100}$
  can be computed by multiplying (mod 100) and (mod 100)

- $2^{23} \pmod{100}$
  can be computed by multiplying (mod 100) and (mod 100)

- $2^{72} \pmod{100}$
  can be computed by multiplying (mod 100) and (mod 100)

2. Now, what patterns have you noticed? Describe three patterns/tricks/ideas:

   (a)

   (b)

   (c)

3. Now, your challenge is to compute the following quantity with the fewest multiplications. Give the steps in a diagram or list.

$$2^{148} \pmod{100}$$

4. Compare to your groupmates. He/she who found the shortest path to the answer wins!

5. Next, the goal is to come up with the most efficient algorithm to compute exponents in general. Work together as a group. You can give an algorithm on paper, in pseudocode, or you can program an actual program in Sage, depending on your coding skills.

   Input: modulus n, base b, exponent k

   Output: A recipe of steps to compute $b^k \pmod{n}$. A step consists of multiplying together two previously computed values. The code should show each step of the multiplication to the screen, and report the total number of steps.

   Example Input: modulus $n = 100$, base $b = 2$, exponent $k = 5$.

   Example Corresponding Output:

   (a) $2 \pmod{100}$ times $2 \pmod{100}$ is $2^2 \pmod{100}$.

   (b) $2^2 \pmod{100}$ times $2^2 \pmod{100}$ is $2^4 \pmod{100}$.

   (c) $2^4 \pmod{100}$ times $2 \pmod{100}$ is $2^5 \pmod{100}$.

   Total steps: 3.

   Once all teams have created their programs, a series of challenges will be supplied and tested and the most efficient algorithm (fewest steps) wins.